SMALL DELAY TRACING DEFECT TESTING

Lakshmaiah Alluri[1] and Hemant Jeevan Magadum[2]

[1]HDG, CDAC, Thiruvananthapuram, India
[2]ITNS, CDAC, Thiruvananthapuram, India

ABSTRACT

This Small Delay Tracing Defect Testing detect small delay defects by creating internal signal races. The races are created by launching transitions along simultaneous two paths, a reference path and a test path. The arrival times of the transitions on a 'convergence' or common gate determine the result of the race. On the output of the convergence gate, a static hazard created by a small delay defect presence on the test path which is directed to the input of a scan-latch. A glitch detector is added to the scan latch which records the presence or absence of the glitch.

KEYWORDS

Delay, Defect, detector, glitch, testing.

1. INTRODUCTION

The traditional delay defect detection techniques having their advantages and disadvantages. In structured test approach, ATPG techniques used to generate input test vector sequences to target classified faults in the circuit. Typically ATPG works in three main phases which are excite the target fault, propagate the fault effect to the observation point through an identified path and justify the values of off-inputs without causing a contradiction. To achieve a desired test coverage, a set of patterns generated by ATPG. The percentage of total number of faults being detected using the generated pattern set out of initial target fault list is called as Test coverage. ATPG targets the faults in two main steps which are generating fault-oriented test patterns and performing fault simulation to determine a list of faults being detected using the generated test patterns. The progressive ATPG tools change these 2 steps into one operation whereas pattern generation. The ATPG generates the test patterns and enables the ATE in order to load pattern data into a chip's scan cells. In order to achieve 100% controllability. Next, these flip-flops are stitched into a single chain or multiple chains based on the number of pads available on the device to perform test. The stuck-at fault model is assumed to be the most common fault model while performing fault simulation. It is being used because of its effectiveness in finding many common defect types. The physical struck at defects can be traced the struck at model capture the struck at zero and struck at one faults by implementing the traditional methods such as Transition delay fault (TDF) , Path-based and Segment delay fault based on ATPG. As late as possible transition fault (ALAPTF) based ATP, N-detect transition fault based ATPG, Timing-aware ATPG.

2. NEED OF SMALL DELAY DEFECT TESTING

This method is for the detection of small delay defects that minimizes the number of delay tests that need to be applied at faster than the rated clock speed. The method requires the longest path to every primary output and scan-latch input (endpoint) to be validated using delay tests or a

David C. Wyld et al. (Eds): VLSIA, INWES, CSIP, BMLI - 2021
pp. 01-06, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.112101

reference path test structure. These longest paths are the reference paths. Once these paths are validated, they are used to refer upper bound delay paths. This process detects small delay defects that occur on the reference paths and on the path segments driving the off-path inputs. The process of testing the off-path segments for little delay defects involves simultaneously propagating signals along both the off-path segment and reference path segment. The gate that serves as the endpoint of both path segments is called the convergence gate. The off-path segment features a smaller delay or a delay adequate to the reference segment delay in normal condition. Otherwise, the reference path is not the longest path, and its transition arrives at the convergence gate before the reference segment signal. If the off-path segment features a small delay defect, the other may occur. The transitions on the inputs to the convergence gate are controlled such that if later a glitch is produced on the output of the convergence gate. A glitch detector, placed at some extent beyond the convergence gate, is employed to capture the glitch, thereby recording the results of the signal race. If a small delay fault occurred, the glitch detector value is scanned out for inspection to determine.

3. ARCHITECTURE OF SMALL DELAY DEFECT TESTING

It makes use of internal signal races along path segments as a way of upper bounding the propagation delay of a test path segment against the delay of reference path segment. So, the upper bounds on the delay of a set of reference paths are determined by using a reference path test structure or by applying standard delay tests. The longest path to every endpoint (scan latch input or primary output) is chosen because the reference path. Figure 2 shows a reference path with multiple endpoints. The longer path to endpoint D is labeled pr for path reference. A transition is launched from a PI or scan-latch A and is captured at endpoint D. If the signal propagates to D within the launch-capture cycle time, then its delay is upper bounded by that point. This process confirms that the reference path does not have a delay defect. The same process is applied to the remaining reference paths. The validated reference paths are then used to bound the delays of other, shorter paths in the circuit. Figure 3 illustrates how this is accomplished. Two path segments are identified as sr (for segment reference) and st (for segment test). Below Figure 1 shows automation flow of methodology.

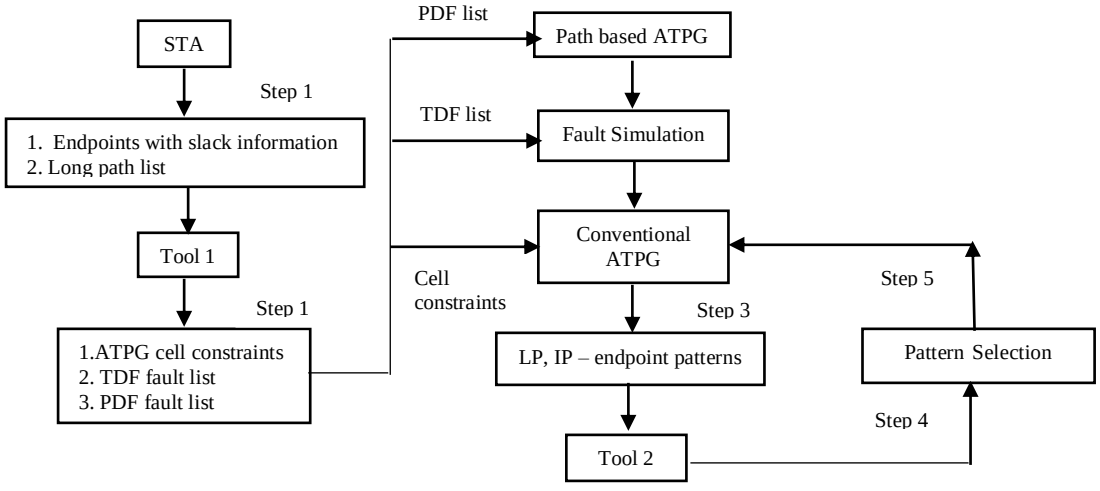

Figure 1. Automation Flow of Methodology

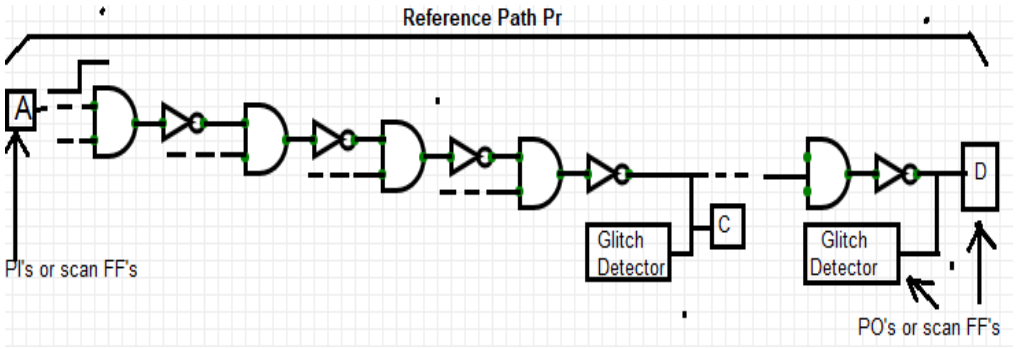

Figure 2. Reference path is longest path to end point

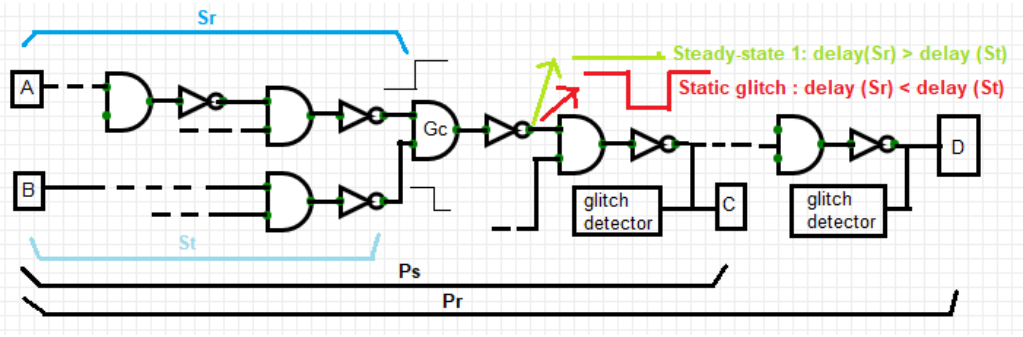

Figure 3. A signal race is created between reference path segment and test path segment driving an off path input to convergence gate.

It launch points labelled A and B, respectively. The endpoints of the segments drive the inputs of a convergence gate, GC. Since the reference path is the longest path to endpoint D, the delay of sr, delay (sr) is greater than or equal to delay(st) by design. The transitions shown at the inputs of GC, i.e., 0 → 1 for sr and 1 → 0 for st cause the output of GC to behave in one of two ways. If delay (sr) >= delay (st) then the 0 along st arrives before the 1 on sr, and the output remains steady-state high. If the opposite is true, i.e., delay(sr) < delay(st), then GC's output switches momentarily with duration proportional to the difference in delays along the two segments.

The relative timing of the two segments is reflected in the output behaviour of GC. One way to record the output behaviour of GC for subsequent inspection is to monitor the state of the path segment between the convergence gate and an endpoint using a glitch detector. If a transition occurs on its input, the glitch detector is designed with a memory element that flips state. Figures 1 and 2 show two glitch detectors at endpoints C and D. For this test, it is possible to use the glitch detector at endpoint D, selecting an endpoint that is closest to the convergence gate. The glitch detector at endpoint C is better for several reasons. First, differences in pull-up and pull-down strengths of gates along a path can compress the width of the glitch (and even eliminate it), making it more difficult or impossible to detect. Second, hazards produced on off-path inputs between the convergence gate and the endpoint may invalidate the result. So keeping this segment small helps minimize these effects. The example shows the reference segment input to GC changing from the dominate value (0) to the non-dominate value (1), and the test segment input transitioning in the opposite direction. Reversing these transitions allows the relative segment delay of the opposite transition along these segments to be tested. for instance , assume the reference segment transition is 1 -> 0 and therefore the test segment transition is 0 -> 1. If the reference segment transition is slower, i.e., delay (sr) > delay (st) then a static hazard is produced. Therefore the reference path segment delay must be larger than the test path segment delay.

4. GLITCH DETECTOR

As indicated above, the width of the static hazard is proportional to the quantity of additional delay introduced by a little delay defect. Therefore, the planning and layout of the glitch detector must be optimized to detect narrow glitches so as to maximise the sensitivity of the tactic to small delay defects. A second criterion is to stay it small, to attenuate the overhead related to the tactic. Last, the glitch detector must be compatible with launch-off-capture and launch-off-shift delay test methodologies so as for it to be deemed practical. One possible implementation of a glitch detector that meets these criteria is shown in Figure 3. The capture-flop input is shown at the highest of the figure and therefore the capture-flop (with scan) is shown on the proper. The remaining gates constitute elements of the glitch detector. An XOR glitch rectifier as shown in fig consists of two inverters is one of the XOR input. The output of the XOR glitch rectifier fed to the one of the input of the latch circut and the other input that is glitch enable signal is fed to the latch. The output of the latch enables the capture flop through MUX. The capture flop output fed to the next scan flop. Example transitions are shown within the figure. The XOR A output drives the input of a latch, i.e., two NOR gates configured with feedback. The output state of the latch is initialized to 0 before conducting the test by setting glitch_en high. The glitch_en control signal is then set low before application of the delay test patterns. If a static hazard is propagated to the capture-flop input as a results of the test, the first rising edge produced by XOR A flips the state of the latch and generates a 1 on the output of NOR A. The result is stored within the latch than the XOR B is enabled the resultant into scan chain mode. If the output of the latch is 0 then the content of the scan chain remains unchanged.. If the latch output is 1, XOR B flips the state of the bit getting into the capture-flop. The results of this test, also because the results of other tests performed simultaneously on other paths, are scanned out after setting glitch_en to 1. Figure 4 shows Glitch detector design.

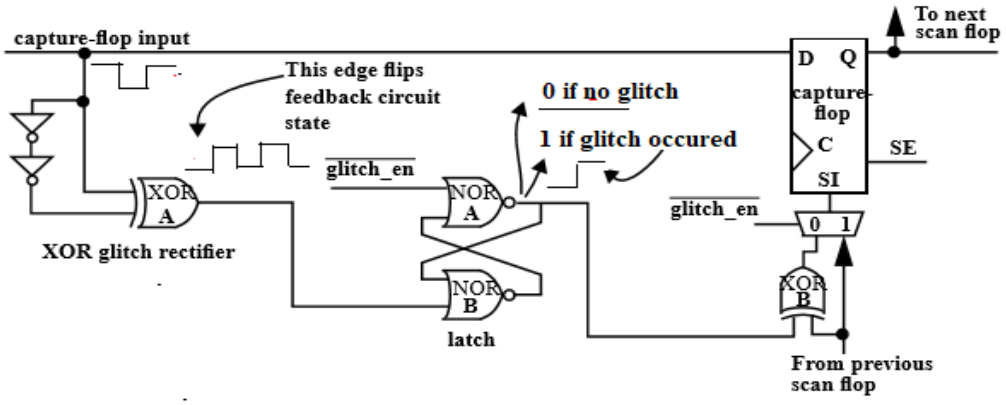

Figure 4. Glitch Detector Design

5. REFERENCE PATH

Reference Path trail sensitization relies on a group of reference paths that are validated to be freed from small delay defects. The reference paths are defined because the longest paths that drive each endpoint (capture latch or primary output). Since all other paths to the endpoints are shorter by definition, it follows that identifying and validation.

An inverter can be inserted in series with this connection as a means of reducing the load to one inverter input, in case of the load capacitance of the inverter and XOR gate is a concern. The coverage of small delay defects in the rest of the circuit can be maximized because of longest

path. The validation of the reference paths can be carried out in one of two ways. An at-speed delay test can be applied to check that each reference path is upper-bounded by the clock cycle time. Although this approach works for the reference paths that are also critical paths in the circuit, it cannot be used to confirm a tight upper timing bound for the shorter reference paths. This is true because the shorter reference paths may have significant slack when tested with an at-speed clock. A straightforward solution for testing the shorter reference paths is to use a faster-than-at-speed clock. Unfortunately, testing results in yield loss due to IR drops and other noise effects because of testing at faster-than-at-speed. Our approach avoids the drawbacks associated with the application of a faster-than-at-speed clock by incorporating a special reference path on chip. Figure 5 shows the proposed test structure, subsequently referred to as the reference path test structure (RPTS). It consists of a launch-flop with scan on the left (enabling standard launch-on-shift/launch-on-capture transition testing), a tri-stateable inverter followed by a string of inverters that form a delay chain. The chain is tapped at each successive inverter output using a MUX. The tap point selected determines the delay from the launch-flop to the input of AND followed by NOT A shown on the right in the figure. The MUX delay select inputs control the selection of the tap point, are controlled by the test engineer using a scan chain. The AND followed by NOT A gate serves the role of the convergence gate, GC, described earlier. The bottom input of the AND followed by NOT A is driven by a path-select MUX. The inputs to the MUX are connected to those endpoints that require a faster-than-at-speed validation. The output of AND followed by NOT A drives a glitch detector and is identical to the glitch detector described in reference to Figure 4. While launching a transition from the launch-flop of the RPTS simultaneously, the shorter reference paths can be validated by applying a delay test to them. The expected delay of the reference path is emulated in the RPTS by selecting the appropriate tap point in the inverter chain. The output of the RPTS's glitch detector reflected with result of the race of the transitions along both paths. For example, with the RPTS configured with a delay larger than the reference path under test, the absence of a glitch indicates the reference path under test is shorter and free of small delay defects. The capture flop for inspection off-chip recorded as result. Accurate delay emulation by the RPTS requires knowledge of the actual delay of its delay chain, which can be obtained through calibration. Calibration is performed by configuring the delay chain into a ring oscillator. This is accomplished by setting RO_en to 1 in Figure 4. The delay select inputs to the MUX are configured so that the entire chain of inverters are part of the ring oscillator. A frequency divider (right side of Figure 4) is used to drive an off-chip pin connected to a frequency measuring instrument. The delay of the chain is the inverse of the measured frequency scaled by the value of the frequency divider. It becomes possible to configure specific delays into the RPTS for validating each of the shorter reference paths, when the RPTS is calibrated.

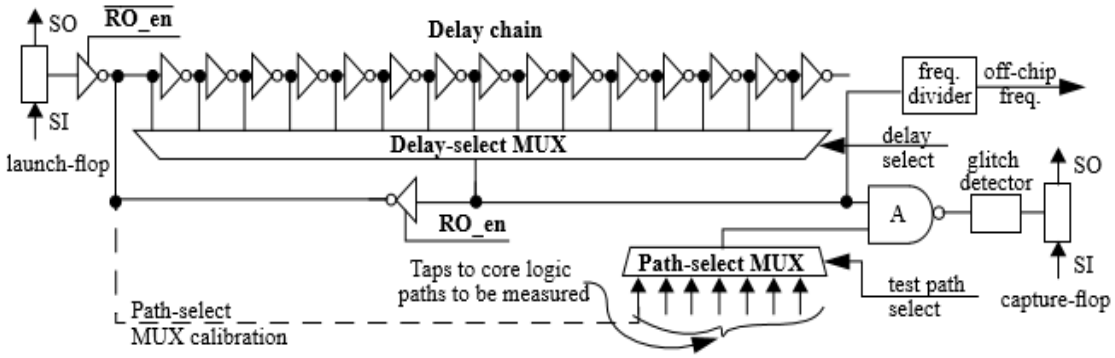

Figure 5. Reference path infrastructure

6. CONCLUSIONS

The advantages of the small delay defect testing include the elimination of a capture clock cycle, which significantly reduces test power issues. Without applying a faster-than-at-speed clock, this method can also detect very small delay defects. It describe a test method that is able to detect very small delay defects without requiring a faster-than-at-speed clock. The strategy also reduces test power by eliminating the capture clock cycle associated with standard delay testing. The technique uses of internal races as a means of bounding the delay of one path segment against another. The result of the test either causes a static hazard to be generated or the transition along a reference path to be halted. Glitch detectors are added to path endpoints as a means of distinguishing these two conditions. A reference path test structure is proposed to validate shorter reference paths against small delay defects. For timing analysis and to measure path delays for validation and debugging of first silicon, this test structure can also be used to aid with correlating models with actual hardware.

ACKNOWLEDGEMENTS

I wish to acknowledge the following colleagues Hardware design group and Intelligent Transportation and networking section who provided their valuable support and guidance to write the paper.

REFERENCES

[1] B. Kruseman, A. K. Majhi, G. Gronthoud and S. Eichenberger, (2004) “On hazard-free patterns for fine-delay fault testing”, ITC , 2004, pp. 213-222.
[2] Jubayer Mahmod, Spencer Millican, Ujjwal Guin, and Vishwani Agrawal, (2019) “Delay Fault Testing: Present and Future”, IEEE 37th VLSI Test Symposium (VTS).
[3] P. Nigh, (2003) “The Increasing Importance of On-line Testing to Ensure High-Reliability Products”, ITC, 2003.
[4] P. Gupta and M. S. Hsiao, (2004) “ALAPTF: A New Transition Fault Model and the ATPG Algorithm”, ITC, 2004.
[5] N. Ahmed, M. Tehranipoor and V. Jayaram, (2006) “Timing-Based Delay Test for Screening Small Delay Defects”, DAC, 2006.
[6] A. D. Singh and G. Xu, (2006) “Output Hazard-Free Transition Tests for Silicon Calibrated Scan Based Delay Testing”, VTS, 2006, pp. 349-355.
[7] B. N. Lee, L. C. Wang and M. S. Abadir, (2004) “Reducing Pattern Delay Variations for Screening Frequency Dependent Defects”, VTS, 2005, pp. 153-160.